Initiation à l'algorithmique et programmation

L3 2024-2025

Travaux Pratiques 4

Voici quelques exercices qui permettront de pratiquer `while` :

Exercice 1. *Recherche de l'inconnue*

Écrivez un programme Python qui demande à l'utilisateur de deviner un nombre entier secret que vous, en tant que programmeur, définissez dans le code. Le programme donnera des indices à l'utilisateur pour l'aider à deviner (par exemple, "trop grand" ou "trop petit"), et le programme s'arrête lorsque l'utilisateur trouve le bon nombre.

Consignes :

- Le programmeur choisit un nombre fixe (par exemple, 42), que l'utilisateur doit deviner.
- Le programme demande à l'utilisateur de faire des propositions jusqu'à ce qu'il trouve le bon nombre.
- Pour chaque tentative, le programme indique si le nombre proposé est "trop grand" ou "trop petit".
- L'utilisateur doit pouvoir voir combien d'essais il a faits avant de deviner le bon nombre.

Exemple :

```
Essayez de deviner le nombre mystère !
Devine : 20
Trop petit !
Devine : 50
Trop grand !
Devine : 42
Bravo ! Tu as trouvé le nombre en 3 essais.
```

Exercice 2. *Compte à rebours*

Écrivez un programme qui prend un entier en entrée et effectue un compte à rebours jusqu'à zéro – imprimez la valeur à chaque étape.

Exemple :

```
Entrée : 5
Sortie :
5
4
3
2
1
0
```

Exercice 3. *Nombres pairs*

Écrivez un programme qui prend un nombre entier en entrée et affiche tous les nombres pairs à partir de 0 jusqu'à ce nombre.

Exemple :

```
Entrée : 8
Sortie :
0
2
4
6
8
```

Exercice 4. *Triangle 1*

Ecrire un programme pour afficher le motif comme un triangle à angle droit en utilisant un astérisque.

Exemple :

```
nbr_lignes = 5
*
**
***
****
*****
```

Instructions :

1. Déclarez une variable entière `nbr_lignes` et demandez à l'utilisateur de saisir lui même la valeur (par exemple, 5).
2. Utilisez des boucles pour afficher le motif de triangle. Vous pouvez utiliser une boucle `while` pour gérer le nombre de lignes et une autre pour le nombre d'astérisques par ligne, ou utiliser simplement une boucle unique.
3. Testez votre programme avec différentes valeurs de pour vérifier que l'affichage est correct pour chaque cas.
4. Commentez votre code pour expliquer les différentes étapes de votre solution.

Exercice 5. *Triangle 2*

Modifiez votre programme pour qu'il affiche un triangle inversé. Le nombre de lignes du triangle est toujours déterminé par `nbr_lignes`.

Exemple :

$\texttt{nbr}_l ignes = 5$

```
*****
****
***
**
*
```